

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Daniel J. Magenheimer	§	Art Unit:	2195
		§		
Serial No.:	10/784,065	§	Confirmation No.:	2613
		§		
Filed:	February 20, 2004	§	Examiner:	Abdullah Al Kawsar
		§		
For:	Flexible Operating System	§	Atty. Dkt. No.:	200315952-1
	Operable as Either Native or	§		(HPC.0500US)
	as Virtualized	§		

Mail Stop Appeal Brief-Patents

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

APPEAL BRIEF PURSUANT TO 37 C.F.R § 41.37

Sir:

The final rejection of claims 1, 3-27, 29-31, 33 and 35-43 is hereby appealed.

I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company, LP. The Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 11445 Compaq Center Drive West, Houston, TX 77707, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF THE CLAIMS

Claims 1, 3-27, 29-31, 33 and 35-43 have been finally rejected and are the subject of this appeal. Claims 2, 28, 32, and 34 have been cancelled.

IV. STATUS OF AMENDMENTS

No amendment after the final rejection of January 14, 2009 has been submitted. Therefore, all amendments have been entered.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the independent claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

Independent claim 1 recites a computer system comprising:

at least one processor (Fig. 7:701; Spec., p. 21, ¶ [0067], ln. 1-8); and

a flexible operating system executable on the at least one processor to (Spec., p. 4, ¶ [0016], ln. 1-11; p. 20, ¶ [0066], ln. 1-9):

determine whether said flexible operating system is being used as a native operating system or as a virtualized operating system on said computer system (Spec., p. 4, ¶ [0016], ln. 3-11); and

execute in a first manner as a native operating system on the computer system in response to detecting that said flexible operating system is being used as the native operation system, and execute in a second manner as a virtualized operating system on said computer system in response to detecting that said flexible operating system is being used as the virtualized operating system (Spec., p. 5, ¶ [0017], ln. 1-13; p. 5, ¶ [0019], ln. 1-16; p. 6, ¶ [0021], ln. 1 - p. 8, ¶ [0025], ln. 18),

wherein said flexible operating system is configured to operate in a non-virtualized environment when said flexible operating system is being used as the native operating system, and is configured to operate in a virtualized environment when said flexible operating system is being used as the virtualized operating system (Spec., p. 5, ¶ [0017], ln. 1-13; p. 5, ¶ [0019], ln. 1-16; p. 6, ¶ [0021], ln. 1 - p. 8, ¶ [0025], ln. 18).

Independent claim 11 recites a method comprising:

implementing at least one operating system on a computer system (Fig. 7:500; Spec., p. 4, ¶ [0016], ln. 1-11; p. 20, ¶ [0066], ln. 1-9);

determining, by said computer system, whether said at least one operating system is a native operating system or a guest operating system on a virtual machine (Spec., p. 4, ¶ [0016], ln. 3-11);

said at least one operating system operating in a first manner if determined that it is a native operating system, wherein the native operating system operates in a non-virtualized environment (Spec., p. 5, ¶ [0017], ln. 1-13; p. 5, ¶ [0019], ln. 1-16; p. 6, ¶ [0021], ln. 1 - p. 8, ¶ [0025], ln. 18); and

said at least one operating system operating in a second manner if determined that it is a guest operating system on a virtual machine, wherein the guest operating system operates in a virtual environment provided by the virtual machine (Spec., p. 5, ¶ [0017], ln. 1-13; p. 5, ¶ [0019], ln. 1-16; p. 6, ¶ [0021], ln. 1 - p. 8, ¶ [0025], ln. 18).

Independent claim 17 recites a computer system comprising:

at least one processor (Fig. 7:701; Spec., p. 21, ¶ [0067], ln. 1-8);

a virtual machine monitor (VMM) (Fig. 1:12; Spec., p. 5, ¶ [0018], ln. 1-13); and

an operating system executable on the at least one processor to (Spec., p. 4, ¶ [0016], ln. 1-11; p. 20, ¶ [0066], ln. 1-9):

determine whether said operating system is running as a virtualized operating system or a native operating system (Spec., p. 4, ¶ [0016], ln. 3-11); and

adapt operation of said operating system depending on whether it is running as the virtualized operating system or native operating system, wherein the native operating system manages hardware resources in a non-virtualized environment without the VMM, and wherein the virtualized operating system manages hardware resources using the VMM (Spec., p. 5, ¶ [0017], ln. 1-13; p. 5, ¶ [0019], ln. 1-16; p. 6, ¶ [0021], ln. 1 - p. 8, ¶ [0025], ln. 18).

Independent claim 27 recites a system comprising:

hardware resources (Fig. 1:13; Spec., p. 5, ¶ [0017], ln. 1-13);

a virtual machine monitor (VMM) (Fig. 1:12; Spec., p. 5, ¶ [0018], ln. 1-13); and

at least one operating system for managing said hardware resources, wherein said at least one operating system is operable to determine whether it is running in a virtualized environment or in a native, non-virtualized environment, wherein said at least one operating system is operable to selectively execute in a first manner if determined that said at least one operating system is running in the native environment and in said second manner if determined that said at least one operating system is running in the virtualized environment, wherein in the first manner said at least one operating system manages said hardware resources without using the VMM, and wherein in the second manner said at least one operating system manages said hardware resources using the VMM (Spec., p. 4, ¶ [0016], ln. 1-11; p. 5, ¶ [0017], ln. 1-13; p. 5, ¶ [0019], ln. 1-16; p. 6, ¶ [0021], ln. 1 - p. 8, ¶ [0025], ln. 18).

Claim 35, set forth below, includes means-plus-function elements, which are identified as required by 37 C.F.R. § 41.37. For each means-plus-function element, the structure, material, or acts described in the Specification as corresponding to each claimed function is set forth by reference to page and line number, and to the drawings, by reference characters.

Independent claim 35 recites a system comprising:

at least one processor (Fig. 7:701; Spec., p. 21, ¶ [0067], ln. 1-8);

a flexible operating system (Spec., p. 4, ¶ [0016], ln. 1-11; p. 20, ¶ [0066], ln. 1-9) executable on the at least one processor and that is capable of acting as either a native operating system or as a virtualized operating system; and

means (Fig. 1:OS) for determining whether the flexible operating system is to be used as a native operating system in a non-virtualized environment without a Virtual Machine Monitor (VMM) or as a virtualized operating system in a virtualized environment with the VMM, wherein the determining means stores information that is accessible by the flexible operating system to indicate whether the flexible operating system is being used as a native or as a virtualized operating system (Spec., p. 5, ¶ [0017], ln. 1-13; p. 5, ¶ [0019], ln. 1-16; p. 6, ¶ [0021], ln. 1 - p. 8, ¶ [0025], ln. 18).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

- A. Claims 1, 3-8, 10-27, 29-31, 33, 35, 37-43 were rejected under 35 U.S.C. § 102(e) as anticipated by U.S. Patent Publication No. 2004/011732 (Bennett).**
- B. Claim 9 was rejected under 35 U.S.C. § 103(a) as unpatentable over Bennett in view of U.S. Patent No. 6,725,289 (Waldspurger).**
- C. Claim 36 was rejected under 35 U.S.C. § 103(a) as unpatentable over Bennett in view of U.S. Patent No. 6,199,159 (Fish).**

VII. ARGUMENT

The claims do not stand or fall together. Instead, Appellant presents separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-headings as required by 37 C.F.R. § 41.37(c)(1)(vii).

- A. Claims 1, 3-8, 10-27, 29-31, 33, 35, 37-43 were rejected under 35 U.S.C. § 102(e) as anticipated by U.S. Patent Publication No. 2004/011732 (Bennett).**

- 1. Claims 1, 3, 4, 8, 10, 43.**

Independent claim 1 recites the following:

- execute in a first manner as a native operating system on the computer system in response to detecting that said flexible operating system is being used as the native operation system, and execute in a second manner as a virtualized operating system on said computer system in response to detecting that said flexible operating system is being used as the virtualized operating system,
- wherein said flexible operating system is configured to operate in a non-virtualized environment when said flexible operating system is being used as the native operating system, and is configured to operate in a virtualized environment when said flexible operating system is being used as the virtualized operating system.

In contrast, as best seen in Fig. 1 of Bennett, both operating systems (OS#1 and OS#2) operate in a respective virtualized environment, as provided by respective virtual machine abstraction 1 and virtual machine abstraction 2 and by the virtual machine monitor (VMM). As

specifically taught by Bennett, both the operating systems depicted in Fig. 1 are guest operating systems. *See, e.g.,* Bennett, ¶ [0019]. By definition, a guest operating system runs within a virtual machine, and therefore, **cannot** constitute a native operating system that operates in a **non-virtualized** environment.

The handling of privileged or non-privileged interrupts by guest software directly or by the VMM, as taught by the passages of Bennett cited by the Office Action, does not change the fact that each guest operating system depicted in Fig. 1 of Bennett operates in a virtualized environment, and cannot be considered to be a native operating system that operates in a non-virtualized environment.

As purportedly disclosing the “wherein” clause of claim 1, the Examiner cited ¶¶ [0024], [0026], and [0027] of Bennett. The cited passages of Bennett refer to processing an interrupt. As explained in ¶ [0026], if an interrupt is generated during operation of guest software, such as a guest OS, then an appropriate interrupt control indicator is consulted to determine whether the interrupt is to be managed by the guest software. If so, the interrupt will be managed by the guest software. On the other hand, as further explained in ¶¶ [0027], if the interrupt is not to be handled by the guest software, then the interrupt is managed by the VMM shown in Bennett.

The determination of whether an interrupt is to be handled by a guest OS or a VMM does not provide any teaching or hint of a flexible operating system that is configured to operate in a non-virtualized environment when the flexible operating system is being used as a native operating system, and configured to operate in a virtualized environment when the flexible operating system is being used as a virtualized operating system. The cited passages of Bennett in ¶¶ [0024], [0026], and [0027] refer to selection between two different entities (guest OS and VMM) to process an interrupt based on a control indicator. These passages provide no hint of a

flexible operating system that is configured to operate in one of a non-virtualized environment or virtualized environment depending upon whether the operating system is to be used as a native or virtualized operating system.

The Examiner further argued that the “claim limitation is broad and [does] not specify how and where the identification is made and what will constitute the machine running in virtual mode or non-virtual mode.” 1/14/2009 Office Action at 11. Moreover, the Examiner argued that the “claim limitation does not specifically disclose the execution state of the different operation or the execution layer that differentiates between the virtual and non-virtual operating mode.” *Id.*

The above statements made by the Examiner are irrelevant to issues of whether or not Bennett discloses the subject matter of claim 1. As specifically shown in Bennett, two guest operating systems 104 and 106 are run in corresponding virtual machine abstractions 102 and 114, as shown in Fig. 1 of Bennett. Thus, both guest operating systems 104 and 106 run in corresponding virtual environments. There is absolutely no teaching or hint given in Bennett that either the operating system 104 or operating system 106 is a flexible operating system that can be used as a native operating system that operates in a non-virtualized environment, or a virtualized operating system that operates in a virtualized environment. The guest operating system 104 and 106 run only in virtualized environments.

The Response to Arguments section in ¶ 43 of the Office Action further pointed to a virtual machine and a VMM mentioned in Bennett. 1/14/2009 Office Action at 11-12. The Examiner argued that the VMM operates in native mode, and the virtual machine is an abstraction layer that runs in virtual mode on the VMM. *Id.* The above statement, even if true, does not provide any hint of the subject matter of claim 1, which is directed to a flexible

operating system that is configured to operate in a non-virtualized environment when the flexible operating system is being used as the native operating system, and is configured to operate in a virtualized environment when the flexible operating system is being used as a virtualized operating system.

In view of the foregoing, it is clear that claim 1 and its dependent claims are allowable over Bennett.

Reversal of the final rejection of the above claims is respectfully requested.

2. Claims 5-7.

Claim 5 depends from claim 1 and is therefore allowable for at least the same reasons as claim 1. Moreover, claim 5 recites that the flexible operating system determines whether the flexible operating system is being used as a native operating system or virtualized operating system by checking a global variable that indicates whether the flexible operating system is being used as the native operating system or the virtualized operating system on the computer system. With respect to claim 5, the Examiner cited the following passages of Bennett: ¶¶ [0032], [0024]. Paragraph [0032] of Bennett states that a current value of an interrupt control indicator is used to determine whether guest software or the VMM is to manage the interrupt. Consulting an indicator to determine whether a VMM or guest software is to manage an interrupt is completely different from checking a global variable that indicates whether the flexible operating system is to be used as a native operating system or as a virtualized operating system.

Claim 5 and its dependent claims are therefore further allowable over Bennett for the foregoing reason.

Reversal of the final rejection of the above claims is respectfully requested.

3. Claims 11, 12, 14-16.

Independent claim 11 is also not anticipated by Bennett, which fails to disclose determining whether an operating system is a native operating system or a guest operating system on a virtual machine, where the operating system operates in a first manner if determined that it is a native operating system and where the native operating system operates in a non-virtualized environment. Moreover, the operating system operates in a second manner if determined that it is a guest operating system on a virtual machine, where the guest operating system operates in a virtual environment provided by the virtual machine.

As discussed above, all Bennett discloses are guest operating systems that run on corresponding virtual machine environments. There is no concept in Bennett of an operating system that selectively operates as a native operating system and a guest operating system.

Therefore, claim 11 and its dependent claims are clearly allowable over Bennett.

Reversal of the final rejection of the above claims is respectfully requested.

4. Claim 13.

Claim 13 depends from claim 11 (indirectly) and therefore is allowable for at least the same reasons as claim 11. Moreover, claim 13 further recites that the operating system determines whether it is being used as a native operating system or a guest operating system based at least in part on a value of a global variable. There is no such teaching in Bennett, which merely refers to checking a control indicator to determine whether the VMM or guest operating system is to manage an interrupt.

Therefore, claim 13 is further allowable for the foregoing reason.

Reversal of the final rejection of the above claim is respectfully requested.

5. Claims 17, 20-26.

Independent claim 17 is also not anticipated by Bennett, which fails to disclose an operating system that determines whether the operating system is running as a virtualized operating system or a native operating system. Claim 17 further recites that the operating system is to adapt its operation depending on whether the operating system is running as the virtualized operating system or native operating system, where the native operating system manages hardware resources in a non-virtualized environment without the VMM, and the virtualized operating system manages hardware resources using the VMM.

In Bennett, the guest operating systems 104 and 106 execute only in corresponding virtual machine environments 102 and 114. There is no operating system that is configurable to operate in one or two different modes, as recited in claim 17.

Therefore, claim 17 and its dependent claims are clearly allowable over Bennett.

Reversal of the final rejection of the above claims is respectfully requested.

6. Claims 18, 19.

Claim 18 depends from claim 17 and is therefore allowable for at least the same reasons as claim 17. Moreover, claim 18 further recites that the operating system determines whether the operating system is running as the virtualized operating system or the native operating system by checking the value of a global variable. In Bennett, the control indicator that is checked determines whether the VMM or the guest operating system is to manage the interrupt, which is completely different from the subject matter of claim 18.

Therefore, claim 18 and its dependent claim 19 are allowable for the foregoing additional reason.

Reversal of the final rejection of the above claims is respectfully requested.

7. Claims 27, 29-31, 33.

Independent claim 27 is also not anticipated by Bennett, which fails to disclose an operating system that is operable to determine whether it is running in a virtualized environment or in a native, non-virtualized environment. Claim 27 further recites that the operating system is operable to selectively execute in a first manner if determined that the operating system is running in the native environment and in the second manner if determined that the operating system is running in the virtualized environment. In the first manner, the operating system manages the hardware resources without using the VMM, and in the second manner the operating system manages the hardware resources using the VMM.

In Bennett, the guest operating systems 104 and 106 execute only in corresponding virtual environments, and these guest operating systems are not configurable to operate in a native, non-virtualized environment.

Therefore, claim 27 and its dependent claims are clearly allowable over Bennett.

Reversal of the final rejection of the above claims is respectfully requested.

8. Claims 35, 37-42.

Independent claim 35 is also not anticipated by Bennett, which fails to disclose a flexible operating system that is capable of acting as either a native operating system or as a virtualized operating system. Claim 35 further recites means for determining whether the flexible operating system is to be used as a native operating system in a non-virtualized environment without a VMM or as a virtualized operating system in a virtualized environment with the VMM, where the determining means stores information that is accessible by the flexible operating system to indicate whether the flexible operating system is being used as a native or as a virtualized operating system.

In Bennett, the guest operating systems 104 and 106 execute only in the virtualized machine abstractions 102 and 114, with no capability provided to allow either of the guest operating systems to operate in a non-virtualized environment without a VMM.

Therefore, claim 35 and its dependent claims are clearly allowable over Bennett.

Reversal of the final rejection of the above claims is respectfully requested.

B. Claim 9 was rejected under 35 U.S.C. § 103(a) as unpatentable over Bennett in view of U.S. Patent No. 6,725,289 (Waldspurger).

1. Claim 9.

In view of the allowance of base claim 1 over Bennett, it is respectfully submitted that the obviousness rejection of claim 9 over Bennett and Waldspurger has been overcome.

Reversal of the final rejection of the above claim is respectfully requested.

C. Claim 36 was rejected under 35 U.S.C. § 103(a) as unpatentable over Bennett in view of U.S. Patent No. 6,199,159 (Fish).

1. Claim 36.

In view of the allowance of base claim 1 over Bennett, it is respectfully submitted that the obviousness rejection of claim 36 over Bennett and Fish has been overcome.

Reversal of the final rejection of the above claim is respectfully requested.

CONCLUSION

In view of the foregoing, reversal of all final rejections and allowance of all pending claims is respectfully requested.

Respectfully submitted,

Date: June 12, 2009



Dan C. Hu
Registration No. 40,025
TROP, PRUNER & HU, P.C.
1616 South Voss Road, Suite 750
Houston, TX 77057-2631
Telephone: (713) 468-8880
Facsimile: (713) 468-8883

VIII. APPENDIX OF APPEALED CLAIMS

The claims on appeal are:

1 1. A computer system comprising:

2 at least one processor; and

3 a flexible operating system executable on the at least one processor to:

4 determine whether said flexible operating system is being used as a native
5 operating system or as a virtualized operating system on said computer system; and

6 execute in a first manner as a native operating system on the computer system in
7 response to detecting that said flexible operating system is being used as the native operation
8 system, and execute in a second manner as a virtualized operating system on said computer
9 system in response to detecting that said flexible operating system is being used as the
10 virtualized operating system,

11 wherein said flexible operating system is configured to operate in a
12 non-virtualized environment when said flexible operating system is being used as the native
13 operating system, and is configured to operate in a virtualized environment when said flexible
14 operating system is being used as the virtualized operating system.

1 3. The computer system of claim 1 wherein said flexible operating system executing
2 in said second manner comprises said operating system acting as a paravirtualized operating
3 system.

1 4. The computer system of claim 3 wherein said paravirtualized operating system is
2 operable to make a call to a Virtual Machine Monitor (VMM) for performing at least one
3 privileged operation.

5. The computer system of claim 1 wherein said flexible operating system determines whether said flexible operating system is being used as the native operating system or the virtualized operating system by:

checking a global variable that indicates whether said flexible operating system is being used as the native operating system or as the virtualized operating system on said computer system.

6. The computer system of claim 5, wherein said flexible operating system is executable to further:

execute an instruction which, when the flexible operating system is being used as the virtualized operating system, causes a Virtual Machine Monitor (VMM) to set at least one configuration bit to a first value, and when the flexible operating system is being used as the native operating system, causes the VMM to set said at least one configuration bit to a different value.

7. The computer system of claim 6, wherein said flexible operating system is executable to further:

set said global variable based at least in part on the value of said at least one configuration bit after executing said instruction.

8. The computer system of claim 1, wherein said flexible operating system is executable to further:

make a call to a Virtual Machine Monitor (VMM) for performing at least one privileged operation.

9. The computer system of claim 8 wherein making the call to said VMM uses an Application Program Interface (API) defined for said VMM.

10. The computer system of claim 8 wherein making the call to said VMM is used for performing said at least one privileged operation if it is determined that said flexible operating system is being used as virtualized operating system on said computer system.

1 11. A method comprising:
2 implementing at least one operating system on a computer system;
3 determining, by said computer system, whether said at least one operating system is a
4 native operating system or a guest operating system on a virtual machine;
5 said at least one operating system operating in a first manner if determined that it is a
6 native operating system, wherein the native operating system operates in a non-virtualized
7 environment; and
8 said at least one operating system operating in a second manner if determined that it is a
9 guest operating system on a virtual machine, wherein the guest operating system operates in a
10 virtual environment provided by the virtual machine.

1 12. The method of claim 11 wherein said determining comprises:
2 said at least one operating system determining whether it is being used as said native
3 operating system or as said guest operating system on the virtual machine.

1 13. The method of claim 12 wherein said at least one operating system determines
2 whether it is being used as said native operating system or as said guest operating system based
3 at least in part on a value of a global variable.

1 14. The method of claim 11 wherein said first manner comprises said native operating
2 system managing hardware resources of the computer system.

1 15. The method of claim 14 wherein said second manner comprises said guest
2 operating system having access to the computer system hardware resources that are managed by
3 a Virtual Machine Monitor (VMM).

1 16. The method of claim 15 wherein said guest operating system makes, for at least
2 one privileged operation, a call to the VMM.

1 17. A computer system comprising:

2 at least one processor;

3 a virtual machine monitor (VMM); and

4 an operating system executable on the at least one processor to:

5 determine whether said operating system is running as a virtualized operating
6 system or a native operating system; and

7 adapt operation of said operating system depending on whether it is running as the
8 virtualized operating system or native operating system, wherein the native operating system
9 manages hardware resources in a non-virtualized environment without the VMM, and wherein
10 the virtualized operating system manages hardware resources using the VMM.

1 18. The computer system of claim 17 wherein said operating system determines
2 whether said operating system is running as the virtualized operating system or the native
3 operating system by checking the value of a global variable.

1 19. The computer system of claim 18 wherein said operating system checks said
2 value of said global variable before performing certain privileged operations.

1 20. The computer system of claim 17 wherein said operating system performs the
2 determining by determining, before execution of certain privileged instructions, whether said
3 operating system is running as the virtualized operating system or native operating system.

1 21. The computer system of claim 20 wherein said operating systems adapts by if
2 determined that said operating system is running as the virtualized operating system, adapting
3 operation of said operating system in executing said certain privileged instructions.

1 22. The computer system of claim 21 wherein said adapting operation of said
2 operating system in executing said certain privileged instructions comprises:
3 making at least one call to the VMM.

23. The computer system of claim 17 wherein said adapting comprises:
calling the VMM for at least one privileged instruction.

24. The computer system of claim 17 wherein said operating system performs the
determining by executing an instruction which, when the operating system is being used as the
virtualized operating system, causes the VMM to set at least one configuration bit to a first value.

25. The computer system of claim 24 wherein said operating system performs the
determining by further determining whether said operating system is running as the virtualized
operating system or native operating system based at least in part on a determined value of at
least one configuration bit after execution of said instruction.

26. The computer system of claim 24 wherein said operating system performs the
determining by further setting a global variable to a value based at least in part on the value of
said at least one configuration bit after execution of said instruction.

27. A system comprising:
hardware resources;
a virtual machine monitor (VMM); and
at least one operating system for managing said hardware resources, wherein said at least
one operating system is operable to determine whether it is running in a virtualized environment
or in a native, non-virtualized environment, wherein said at least one operating system is
operable to selectively execute in a first manner if determined that said at least one operating
system is running in the native environment and in said second manner if determined that said at
least one operating system is running in the virtualized environment, wherein in the first manner
said at least one operating system manages said hardware resources without using the VMM, and
wherein in the second manner said at least one operating system manages said hardware
resources using the VMM.

29. The system of claim 27 wherein said first manner comprises acting as a native
operating system.

1 30. The system of claim 27 wherein said second manner comprises acting as a
2 paravirtualized operating system.

1 31. The system of claim 30 wherein said paravirtualized operating system is operable
2 to make a call to the VMM for performing at least one privileged operation.

1 33. The system of claim 27 wherein said at least one operating system adapts its
2 operation to make a call to said VMM for performance of at least one privileged instruction
3 when said at least one operating system determines that it is running in a virtualized
4 environment.

1 35. A system comprising:
2 at least one processor;
3 a flexible operating system executable on the at least one processor and that is capable of
4 acting as either a native operating system or as a virtualized operating system; and
5 means for determining whether the flexible operating system is to be used as a native
6 operating system in a non-virtualized environment without a Virtual Machine Monitor (VMM)
7 or as a virtualized operating system in a virtualized environment with the VMM, wherein the
8 determining means stores information that is accessible by the flexible operating system to
9 indicate whether the flexible operating system is being used as a native or as a virtualized
10 operating system.

1 36. The system of claim 35 wherein the determining means makes the determination
2 during a boot-up process of the system.

1 37. The system of claim 35 further comprising: means for virtualizing resources of
2 said system and multiplexing said resources among one or more virtualized operating systems.

1 38. The system of claim 35 wherein said flexible operating system is operable to
2 access the stored information to determine whether said flexible operating system is being used
3 as a native or as a virtualized operating system.

1 39. The system of claim 38 wherein if determined that it is being used as a virtualized
2 operating system, said flexible operating system acting as a virtualized operating system.

1 40. The system of claim 38 wherein if determined that it is being used as a native
2 operating system, said flexible operating system acting in a first manner, and if determined that it
3 is being used as a virtualized operating system, said flexible operating system acting in a second
4 manner.

1 41. The system of claim 40 wherein said first manner comprises acting as a native
2 operating system, and wherein said second manner comprises acting as a paravirtualized
3 operating system.

1 42. The system of claim 35, wherein the virtualized operating system manages
2 hardware resources of the system by using the VMM, and wherein the native operating system
3 manages the hardware resources in the non-virtualized environment without using the VMM.

1 43. The computer system of claim 1, wherein the virtualized operating system
2 manages hardware resources of the system by using a virtual machine monitor (VMM), and
3 wherein the native operating system manages the hardware resources in the non-virtualized
4 environment without using the VMM

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.